

# BINARY PARTITION TREE AS AN EFFICIENT REPRESENTATION FOR FILTERING, SEGMENTATION AND INFORMATION RETRIEVAL

Philippe Salembier and Luis Garrido

Universitat Politècnica de Catalunya  
c/ Gran Capita, s/n, 08034 Barcelona, SPAIN  
Tel: (34) 9 3401 7404, Fax: (34) 9 3401 6447  
E-mail: {philippe,oster}@gps.tsc.upc.es, <http://gps-tsc.upc.es/imatge>

## Abstract

*This paper discusses the interest of Binary Partition Trees as shape-oriented image representations. Binary Partition Trees concentrate in a compact and structured representation a set of meaningful regions that can be extracted from an image. This representation can be used for a large number of processing goals such as filtering, segmentation, information retrieval and visual browsing. Furthermore, the processing of the tree representation leads to very efficient algorithms.*

## 1. INTRODUCTION

This study about Binary Partition Trees intends to contribute to several fields of image processing:

**Connected operators:** these filtering tools [9, 2, 8, 10, 1] are derived from mathematical morphology. They interact with the signal by means of *flat zones* (largest connected components of the space where the image is constant). A connected operator is an operator that only merges flat zones. As it cannot introduce any new contour, it simplifies as well as preserves the contour information. However, one of the major drawbacks of classical operators is that they consider that objects composing the scene are either bright or dark image components (as a result, they simplify either bright or dark objects). This very crude assumption limits their usefulness for certain applications. In this paper, we discuss the interest of Binary Partition Trees to create new connected operators that do not suffer from this restriction.

**Segmentation:** a large number of segmentation techniques such as region growing or watershed rely on iterative merging strategies. These algorithms sequentially merge either pixels or regions. In practice, the class of rules used to control the merging process is restricted. Indeed, rules involving the global optimization of a criterion without specific property (such as increasingness) are difficult to deal with. In such cases, it is difficult to know, at a given time instant, if a particular merging will eventually lead to the optimization of the global criterion and, for practical reasons, it is generally not realistic to memorize and to keep track of all possible merging so that they can be undone in future steps. The Binary Partition Tree approach to segmentation presented in this paper involves two steps: in the first one, a sequence of merging is defined. This sequence can be viewed as a spanning tree of the Region Adjacency Graph where a set

of merging candidates have been selected and ordered. Then, in a second step, some of the merging candidates defined in the first steps are actually activated. The tree structure of the merging allows us to work with fairly complex criteria.

**Content description:** there is currently a strong interest in defining content descriptors for information retrieval (MPEG7 for example). If we consider a low level descriptor such as shape, one of the first issues to be faced is the selection of the resolution scale at which the shape analysis has to be done. Since the queries done during the retrieval may deal with very different scales, it is not pertinent to *a priori* fix the analysis scale. As a result, a shape descriptor should be able to describe shapes at multiple levels of resolution. The Binary Partition Tree possesses this feature of multi-scale shape representation and could be considered as a low-level descriptor of shape information.

The organization of this paper is as follows: Binary Partition Trees are presented in section 2. The application of Binary Partition Trees to information retrieval, segmentation and filtering is respectively discussed in sections 3, 4 and 5. Finally, conclusions are given in section 6.

## 2. BINARY PARTITION TREES

### 2.1. Definition

A Binary Partition Tree is a structured representation of regions that can be obtained from an initial partition. An example is shown in Fig. 1. The tree leaves represent regions that belong to an initial partition shown in Fig. 1.b. The remaining nodes represent regions that are obtained by merging two children regions. The root node represents the entire image support. As can be seen, the tree represents a set of regions at different scales of resolution. It should be considered as a compromise between representation accuracy and processing efficiency. Indeed, all possible merging of regions belonging to the initial partition are not represented in the tree. However, as will be seen in the sequel, the tree representation allows fast and sophisticated processing techniques to be implemented.

### 2.2. Computation of Binary Partition Trees

The Binary Partition Tree should be created in such a way that the most "interesting" or "useful" regions are represented. This issue is, most of the time, application dependent. However, a possible solution, suitable for a large number of cases, is to create the tree by keeping track of the

---

This work has been partially supported by France-Telecom/CCETT under the contract 96ME22.

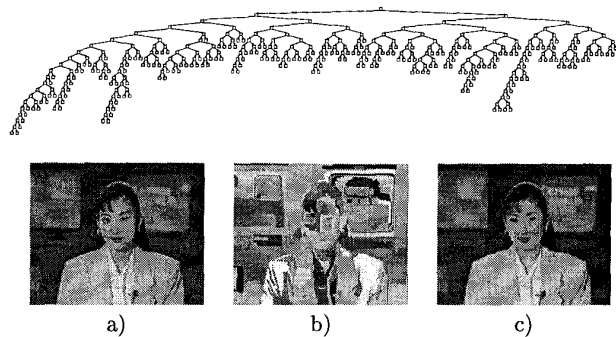


Figure 1: Example of Binary Partition Tree (top). a) original image. b) initial partition with 200 regions. c) regions of the partition represented by their mean value.

merging steps performed by a segmentation algorithm based on region merging (see [3] for example). In the following, this information is called the *merging sequence*. Starting from an initial partition which can be the partition of flat zones or any other pre-computed partition (for example the partition of 1.b), the algorithm merges neighboring regions following a homogeneity criterion until one single region is obtained.

The examples shown in this paper have been computed using a color homogeneity criterion. However, it has to be noticed that the tree creation is not limited to this criterion. For instance, if the original frame belongs to a sequence of images, the homogeneity criterion may be modified during the merging process: in the first phase of the process, regions can be merged based on their similarity in color, whereas, a motion similarity can be used in the second phase. As a result, the Binary Partition Tree can involve regions that are homogeneous either in color (lower tree levels) or in motion (upper tree levels). This approach is not further described in this paper and left for future publications. In the sequel, we assume the tree has been created and we discuss the interest of this representation for various image processing tasks.

### 3. INFORMATION RETRIEVAL

#### 3.1. Detection/Recognition tools

One of the key issue of information retrieval is to be able to efficiently identify objects corresponding to a given query. A direct approach is to evaluate a criterion defining the object(s) of interest on the various regions present in the image. In the sequel, we describe a simple example of circular objects detection but the approach can be used for any criterion that has to be optimized over a set of regions. The circularity is defined as the squared perimeter divided by the area and is minimum for circular objects for which it is equal to  $4\pi$ .

In order to detect the presence of circular objects, the Binary Partition Tree is a particularly attractive representation since it proposes a reduced number of regions which are assumed to be the most homogeneous for different levels of resolutions. An example is shown in Fig. 2. In this case, the initial partition is made of  $N = 100$  regions. As a result,

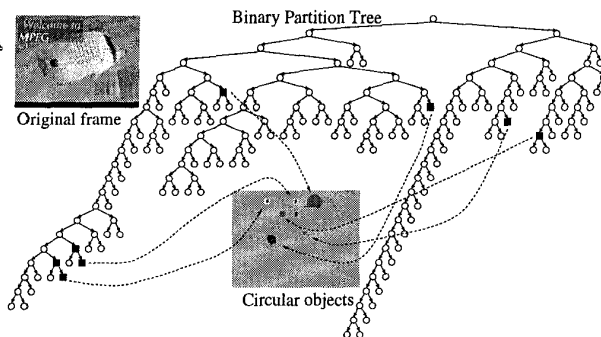


Figure 2: Example of detection of circular objects (initial partition with 100 regions).

the set of regions represented by the Binary Partition Tree equals  $2N - 1 = 199$ . So the algorithm has to measure the circularity of only 199 regions. Fig. 2 presents the tree where circular regions have been represented by a dark square. A spatial representation of these regions can also be seen. They correspond to the letters "o" appearing in "Welcome to the MPEG World" message and to various circular objects.

#### 3.2. Visual browsing

The previous example involves the evaluation and the optimization of a local criterion independently on each region. By contrast, the following browsing example discusses an approach where the optimization is global on the tree structure.

Browsing is an important functionality for information retrieval. Most of the time, the user wants to have a rough idea on the query results. The goal is not to visualize a high quality image, but to simply be able to discard or not the query result. A representation based on Binary Partition Tree is very interesting to deal with such a functionality. Indeed, as shown in [7], partition trees in general are appropriate to define optimum pruning strategy in the rate/distortion sense with restriction on the rate to be transmitted or the distortion of the coded image. The interested reader is referred to [5, 6] for a description of the rate/distortion optimization issue and to [7] for a discussion on how the optimization can be used within the framework of partition trees.

Let us illustrate the approach by an example. Fig. 3 shows a Binary Partition Tree corresponding to a initial partition involving 100 regions. If this original image would have to be transmitted for browsing, and assuming a coding strategy involving the coding of the contours with chain code and of a constant color value for each region, the cost in term of bits would be equal to 14000. However, for visualization purposes, this strategy is not optimum. We show in Fig. 3 three examples of coded images at 3000, 7000 and 11000 bits. As can be seen, the image coded at 11000 bits is visually equal to the initial partition image. In the case where the transmission rate is very low, higher compression factors may be used while allowing the user to have an idea about the image content. Two coding strategies are shown in the tree repre-

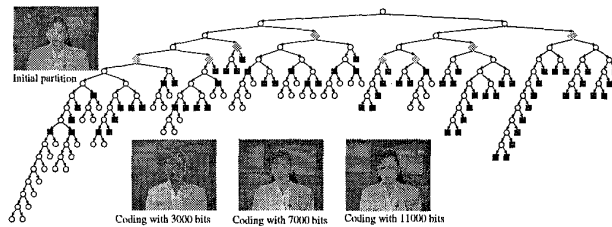


Figure 3: Examples of pruning for visualization: Black squares (gray rhombus) in the tree defines the optimum solution in the rate-distortion sense for 11000 bits (3000 bits).

segmentation. The first one corresponds to the optimum solution for 11000 bits and is shown with dark squares. The second one shown in gray rhombus gives the optimum solution at 3000 bits. As can be seen, for low bit rates, the algorithm selects regions close to the root of the tree. For higher bit rates, a large number of small regions providing details about the image content can be transmitted.

## 4. SEGMENTATION

### 4.1. Direct approach

The Binary Partition Tree representation is particularly suitable to generate segmentation results. Two segmentation strategies are discussed in the following. The first one is a segmentation following a “direct” approach. It consists in merging the regions that are the most similar until a termination criterion is reached. Examples of termination criteria are the number of regions or the Peak Signal to Noise Ratio (PSNR) between the original and the modeled images.

The Binary Partition Tree representation is particularly suitable for the “direct” segmentation approach. Indeed, the merging sequence, which has been used to create the tree, defines the similarity between regions and starts by merging the most similar regions. Therefore, the segmentation can be computed by progressively deactivating the nodes following the merging sequence until the termination criterion is met (required number of regions or PSNR). Using the same initial partition as the one of Fig. 3, we show four examples of segmentation in Fig. 4. In all cases, the termination criterion is defined by number of regions.

### 4.2. Marker & propagation approach

An alternative approach to the direct segmentation is the so-called morphological [4] or “Marker & propagation” approach. The strategy consists, first, in “marking” (defining with markers) the interior of the regions to be segmented and, second, in performing a propagation of these markers to eventually define the regions contours. This second step can be viewed as the definition of the zone of influence of each marker. Let us mention, that depending on the application, the markers can be computed automatically or manually.

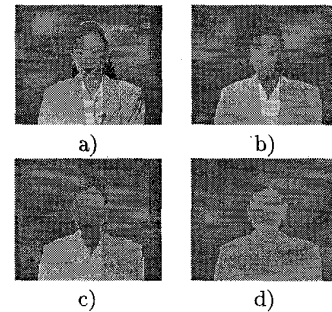


Figure 4: Four examples of “direct” segmentation: a) 50 regions, b) 15 regions, c) 8 regions, d) 2 regions

Propagation processes based on similarity between neighboring regions can be easily implemented in the Binary Partition Tree structure: consider a set of markers corresponding to a set of tree nodes that have to be propagated by merging with neighboring regions. By construction of the Binary Partition Tree, the most similar neighboring region with respect to the marker is represented by the marker’s brother and the result of the merging is represented by the marker’s father. Of course this propagation can only be done if the brother is not in conflict with the marker, that is if none of the brother’s descendant has been assigned to a different marker.

A complete example is shown in Fig. 5. In this example, we assume that a user has defined two markers (dark and gray). The first step is to assign the markers to the leaves of the tree (Fig. 5.top). Then, the propagation process creates three connected components (Fig. 5.bottom). The two first ones correspond to the zones of influence of the markers whereas the last one remains without label. This situation means that the similarity between regions defined by the two markers is higher than any combination with unlabeled regions. The propagation process is controlled in the sense that the algorithm does not blindly assign all regions to markers. Unlabeled regions are produced because they are judged as being “too different”. As can be seen, the face and shoulders regions defined by the markers have been properly segmented and the background has been merged with none of these regions. Finally, note that if necessary, the algorithm could be modified to force a label assignment to all regions.

## 5. FILTERING TOOLS

### 5.1. Pruning strategy

In this section, our objective is to define new connected operators, in particular self-dual operators. First, let us recall the classical strategy used for connected operators. As discussed in [8], the approach involves first the creation of a tree representation of the image (*Max-tree* or *Min-tree*), second the assessment of a criterion for each node of the tree and third the definition of a tree pruning strategy. The pruning defines a new partition and each region is represented by a constant value (minimum in the case of *Max-tree* and maximum in the case of *Min-tree*).

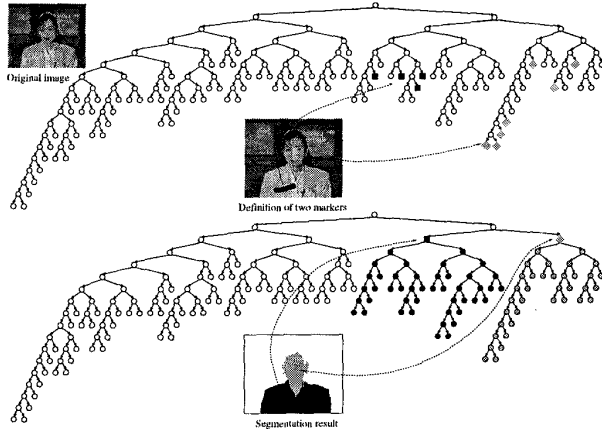


Figure 5: Example of segmentation with marker & propagation strategy. Top: Binary Partition Tree where the leaves intercepted by markers have been indicated. Bottom: Result of the propagation process.

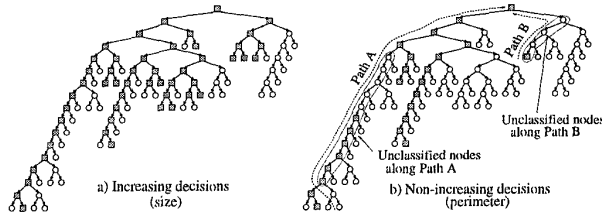


Figure 6: Increasingness of the criterion: a) ex. of increasing criterion: size. b) ex. of nonincreasing criterion: perimeter.

Mathematically, a criterion  $C$  assessed on a region  $X$  is said to be increasing if the following property holds:

$$\forall X_1 \subseteq X_2 \Rightarrow C(X_1) \leq C(X_2) \quad (1)$$

Assume that all nodes corresponding to regions where the criterion value is lower than a given threshold should be removed by merging. Let us define two classes of nodes: the *Remove* class and the *Preserve* class. The *Remove* (*Preserve*) class is made of nodes for which all descendants (ancestors) should be removed (preserved). If the criterion is increasing, all nodes of the tree can be classified into one of the two classes. In this case, the pruning strategy is straightforward: merge all nodes belonging to the *Remove* class. An example of Binary Partition Tree with increasing decision is shown in Fig. 6.a. The criterion used to create this example is the size measured as the number of pixels belonging to the region.

If the criterion is not increasing, the pruning strategy is not straightforward since some nodes cannot be classified in one of the two classes. An example of such criterion is the

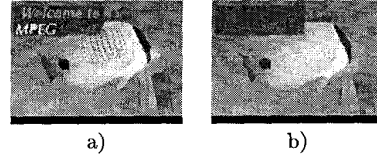


Figure 7: Size-oriented simplification. a) simple size simplification. b) size simplification with propagation strategy

region perimeter. Fig. 6.b illustrates this case. If we follow either Path A or Path B in Fig. 6.b, we see that there are some oscillations of the decisions and some nodes cannot be classified into one of the two classes. In practice, the non-increasingness of the criterion implies a lack of robustness of the operator. For example, similar images may produce quite different results or small modifications of the criterion threshold involve drastic changes on the output. In [8], a similar issue is discussed in the context of *Max-tree* representation for anti-extensive connected operators. The proposed solution consists in applying a transformation on the set of decisions. The transformation should create a set of increasing decisions while preserving as much as possible the decisions defined by the criterion. This problem may be viewed as dynamic programming issue that can be efficiently solved with a Viterbi algorithm. The interested reader is referred to [8] for more details on this issue.

## 5.2. Filtering example

Once the tree has been pruned, the output partition is computed and each region is modeled by a constant value. In the following we assume that the median is used. A first example of size-oriented simplification is shown in Fig. 7.a. The size threshold has been set to 50 pixels. At first glance, this result may be surprising because a large number of regions smaller than 50 pixels are still visible in the filtered image (the texture of the fish for example). To understand this result, let us analyze the Binary Partition Tree shown in the left side of Fig. 8. In this tree, a large number of configurations where the two brothers belongs to two different classes can be seen. Note that since the criterion is increasing, the father of these two nodes has to be classified as *preserve*. In terms of regions, this configuration means that one of the brother as well as the father correspond to large regions whereas the other brother is of small size.

For certain applications, it may be necessary to force the operator to remove these configurations so that all flat zones of the filtered image are guaranteed to fulfill the simplification criterion. This modification can easily be implemented using the propagation process explained in section 4.2. The idea is explained in Fig. 8. The first step consists in defining the markers. These markers are all *Preserve* nodes as well as *Preserve* nodes that have two *Remove* children. In the example of Fig. 8, there are five markers. The second step defines the filtered partition by propagating these markers as in the case of segmentation described in section 4.2. Fig. 8.right shows the result of this propagation on the Binary Partition

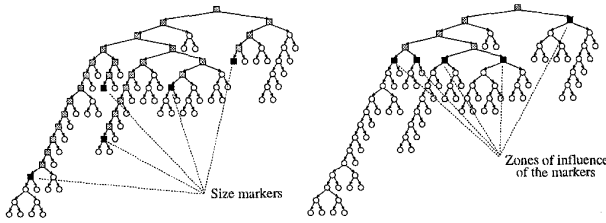


Figure 8: Size-oriented simplification. Left) Binary Partition Tree with size criterion. The black squares indicate the size markers. Right) Definition of the zones of influence of the size markers.

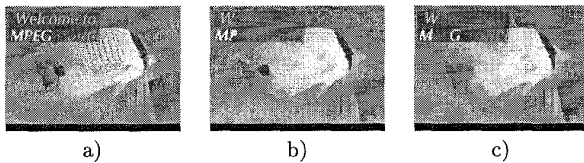


Figure 9: Example of self-dual connected operators. a) original image. b) size simplification. c) perimeter simplification

Tree. A size-oriented simplification of the *Bream* image using this strategy is presented in Fig. 7.b. All regions of size smaller than 50 pixels have been removed.

Fig. 9 illustrates two simplification criteria. Fig. 9.b corresponds to a size criterion whereas Fig. 9.c corresponds to a perimeter criterion. In both cases, the transition regions have been removed by propagation of the markers as explained above. Moreover, the Viterbi algorithm has been used in the case of the perimeter since this criterion is not increasing. The difference between the two simplification criteria can be seen in the simplification of the text appearing in the upper left corner. These filtering tools are self-dual connected operators and generalize the results reported in [8]. They possess the attractive feature of simplifying the image while preserving the contour information.

## 6. CONCLUSIONS

In this paper, we have discussed the interest of Binary Partition Tree representations for several image processing tasks. This representation combines a large number of regions that can be extracted from an image. Although the tree construction was not the main focus of this paper, the use of segmentation algorithms relying on merging techniques has been discussed. Note that this is not the only possibility and top-down or supervised approaches should be investigated. The regions contained in the tree are organized in a hierarchical structure. This organization allows the implementation of fast and sophisticated techniques (the marker propagation of section 4.2 or the Viterbi algorithm of section 5.1).

The processing of the Binary Partition Tree generally

consists in defining which nodes (and corresponding regions) are of interest for a particular image processing task. In this framework, we have discussed examples of minimization of local criteria (circular object detection, section 3.1) as well as the minimization of global criteria (rate/distortion optimization for browsing functionality, section 3.2). The Binary Partition Tree gives access to some (not all) neighborhood as well as similarity relationships between regions. This feature allows the implementation of propagation techniques that are particularly useful for segmentation application (section 4.2). Finally, pruning strategies lead to the definition of new connected operators. In this case, we have seen that the increasingness of the merging criterion is an important issue. In the case of a non-increasing merging criterion, a Viterbi algorithm can be used to define the pruning strategy (section 5.1). Note, that the specific criteria (circularity, size, perimeter, etc) used in this paper are just simple examples that were selected to explain the main issues involved in the representation. For a particular application, more useful and possibly more complex criteria may be used. Let us mention for instance complex shape characteristics, texture features, motion information in the case of sequences, etc. We will investigate such criteria in the future.

## 7. REFERENCES

- [1] E. Breen and R. Jones. An attribute-based approach to mathematical morphology. In *International Symposium on Mathematical Morphology*, pages 41–48, Atlanta (GA), USA, May 1996. Kluwer Academic Pub.
- [2] J. Crespo, J. Serra, and R.W. Schafer. Theoretical aspects of morphological filters by reconstruction. *Signal Processing*, 47(2):201–225, 1995.
- [3] L. Garrido, P. Salembier, and D. Garcia. Extensive operators in partition lattices for image sequence analysis. *EURASIP Signal Processing*, 66(2):157–180, April 1998.
- [4] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, September 1990.
- [5] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. on Image Processing*, 2(2):160–175, April 1993.
- [6] E. Reusens. Joint optimization of representation model and frame segmentation for generic video compression. *EURASIP Signal Processing*, 46(11):105–117, Sep. 1995.
- [7] P. Salembier, et al.. Segmentation-based video coding system allowing the manipulation of objects. *IEEE Trans. on Circuits and Systems for Video Technology*, 7(1):60–74, February 1997.
- [8] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Trans. on Image Processing*, 7(4):555–570, April 1998.
- [9] P. Salembier and J. Serra. Flat zones filtering, connected operators and filters by reconstruction. *IEEE Trans. on Image Processing*, 3(8):1153–1160, August 1995.
- [10] L. Vincent. Morphological gray scale reconstruction in image analysis: Applications and efficient alg. *IEEE Trans. on Image Processing*, 2(2):176–201, April 1993.